AI Magazine Volume 6 Number 2 (1985) (© AAAI)

Knowledge Acquisition from Multiple Experts

Sanjay Mittal

Intelligent Systems Lab, Xerox PARC, Palo Alto, CA 94304

Clive L. Dym

Dept. of Civil Eng., Univ. of Massachusetts, Amherst, MA 01003

Abstract

Expert system projects are often based on collaboration with a single domain expert. This leads to difficulties in judging the suitabilitity of the chosen task and in acquiring the detailed knowledge required to carry out the task This anecdotal article considers some of the advantages of using a diverse collection of domain experts.

In this article we discuss some of issues facing potential knowledge engineers with respect to information they need from their domain experts. These issues deal with both judging the suitability of the task and with acquiring detailed knowledge once the suitability issue has been satisfactorily decided. An impression engendered by many publications (Dixon and Simmons, 1983; Hayes-Roth *et al.*, 1983) in the field is that all you need to build an expert system is one domain expert and a favorite programming language, often LISP, though more recently also PROLOG Experience (McDermott, 1981; Smith and Baker, 1983) with systems which have reached some level of maturity highlights the problems with this rather simplistic view, especially the notion that one expert's rules constitutes expertise To avoid some of the pitfalls of relying on a single expert, we propose an approach based on using multiple experts in a domain.

The approach described here is an empirical one based on our experience with different expert systems. Anecdotes from various projects illustrate the issues. We hope that this article will open up discussion on these issues and enable more rigorous scientific analysis to be carried out in the future In the short term, this article may help prevent some ill-conceived projects or steer them in the right direction.

Judging the Suitability of an Expert Systems Task

Before even beginning to build an expert system, you must decide if the domain is suitable, given the current state-of-the-art of both the technology of knowledge engineering and the art of acquiring knowledge. One of the first problems is finding out who the experts are and what problems they solve.

Matching Experts and Problems

It is obvious that the nature and extent of an expert's expertise be ascertained before an expert system is, or can be, built. But there is some circularity inherent in the process, because before a task can be selected for building a system, experts must be available for questioning about the domain. However, in complex and varied domains such as medicine or engineering, any given expert is often very knowledgeable about only a small subset of the tasks in the domain. Thus there is a potential for a mismatch here in that a single domain expert (or small group of them) may suggest a task as suitable for modeling but

This article is based on work performed in collaboration with many othei colleagues and it is a pleasure to acknowledge their influence on the ideas piesented here The MDX project was a collaboration between the first author, B Chandrasekaran, and J W Smith at Ohio State Some of the key people in the DARN project were Daniel Bobrow, Johann deKleer, and Mark Stefik at Xerox PARC and Milt Mallory and Ron Brown at Xerox OSD. The PRIDE project is an ongoing project involving the authois and Mahesh Moijaiia and George Rollei at Xerox RBG The second author would also like to acknowledge the generous assistance of Xerox in enabling him to spend a sabbatical year at PARC We are grateful for the environment at PARC that enables such inteidisciplinary collaborations We are especially grateful to Mark Stefik for his encouragement and comments Daniel Bobrow and Ken Kahn made helpful comments on eailier drafts The conclusions drawn here do not necessarily reflect those of our other collaboratois oi our institutions

that judgment may itself be erroneous. We suggest that for the task selection process itself, a set of experts may prove more useful.

Experience with the MDX project at Ohio State illustrates the difficulties in relying on a single expert in formulating a suitable problem. When the first author and colleagues at Ohio State were looking for a suitable medical domain in which to explore building a knowledge-based system they approached some contacts in the medical faculty about promising areas. One professor very enthusiastically suggested the area of cholestatic diseases (an important subset of liver diseases) because he believed, quite rightly even in retrospect, that while cholestatic diseases were widely studied and known about in the medical field, there were many problems in correctly diagnosing the cause of cholestasis He further believed that the diagnoses would be helped by making the large amount of medical data available from the various tests accessible in a usable fashion to medical practitioners. This seemed to fulfill many of the criteria for taking a knowledge-based approach to diagnosis and the next couple of years were spent in building a set of expert systems MDX (Chandeskran and Mittal, 1983a), PATREC (Mittal, Chandeskran, and Sticklen, 1984), and RADEX (Chandeskran *et al* , 1980)—which collectively did a pretty decent job (Tatman, 1982) of diagnosing cholestatic diseases. The knowledge base was carefully constructed with the active collaboration of practicing physicians and by consulting textbooks and journal articles.

However, when the system was demonstrated to a group of professors and interns who were involved in the daily practice of managing patients (some of whom were cholestatic), with a view towards enlisting their cooperation in performing trials on the use of the system in a real hospital setting, they were not impressed. Upon questioning them, we realized a fundamental dilemma that must be familiar to other researchers in AI: while the experts acknowledged the diagnostic acumen of the MDX system, they did not find that particularly impressive because they were also doing the same task on a routine basis. *Instead, they were more interested in a computer system which could help in performing tasks with which they had difficulty.* To them, at least, the problem with cholestatic diseases was not diagnostic reasoning *per se*, but rather one of developing medical tests that were more precise in visualizing the liver and the biliary system (which are affected by cholestatic diseases) so they could make a correct diagnosis.

A secondary concern was one of deciding when to order some of the existing imaging tests which, while quite useful, were expensive and invasive. In retrospect, it did not matter which expert's perception of the problem was correct. From the standpoint of application, matching the needs of the intended users was of paramount importance. The irony is that while MDX was never tried in clinical situations, it generated valuable insights about diagnostic reasoning (Chandrasekaran and Mittal, 1983a), knowledge compilation (Chandrasekaran and Mittal, 1983b), and structuring knowledge bases (Chandrasekaran and Mittal, 1983a). Neverthless, it is a valuable lesson, and it is clear that some approach needed to be devised to obtain a better perspective on a problem domain before embarking on building an expert system.

Make sure your experts are practicing experts in the selected task. Furthermore, if you are planning to build an expert system to aid some human experts, find out where they really need help. In other words, identify the intended users in advance of building the system. Note how the problem was exacerbated in the above case by relying on only one expert for task selection

Experience with DARN

The DARN project (Mittal, Bobrow, and deKleer, 1984) at Xerox was an attempt by the first author and other colleagues at Xerox PARC to explore where, and if at all, knowledge-based approaches would help in maintaining computer hardware systems. However, instead of following our favorite approaches to diagnosis we decided to closely examine what different technicians were already doing, where the bottlenecks were, and who might have most to gain from a computerized reasoning system

As a first step, we (the knowledge engineers) talked to as many different people involved in computer maintenance as possible: Technicians in research labs, field technicians, service manual writers, service center people, etc It was soon clear that different people worked on different aspects of the problem:

- Servicing broken machines with a view towards bringing them up quickly
- Pinpointing faulty chips on a circuit board so that the board could be repaired
- Answering calls from the field technician in the middle of a diagnostic session.

There was much in common in terms of a shared understanding of the hardware, software, bugs, and repair strategies. However, each set of experts had separate goals, assumptions, and problem solving approaches. One could almost say that many different kinds of expertise had evolved in what appeared to be a single domain of expertise, *viz.*, maintenance.

Let us illustrate with simple examples. Take the problem of identifying faulty chips on a circuit board. The field technicians, invariably, were *not required* to diagnose a malfunctioning computer down to the broken chip. It was sufficient to identify the malfunctioning board, replace it with a correct one and make sure that the time during which the system was down was minimized. On the other hand, technicians in the research laboratories often did both activities. As a result, they had to take a more rigorous approach to diagnosing the faults; merely making the system operational again was not sufficient. Another aspect of the difference was that the technicians at the research laboratories, by the nature of the environment, could develop different strategies. For example, the availability of many other similar machines at the same site made board-swapping a viable strategy for isolating faults.

By talking to a variety of experts early on in the project it was possible to get a better understanding of the different kinds of expertise prevalent in the domain and which expert practiced which kinds.

Interviewing Multiple Experts.

At the begining of 1984, the second author started a joint project (PRIDE 14,15) between Xerox PARC and Xerox Reprographics Business Group (RBG) to explore building an expert system for the design of some sub-systems of a copier. However, before selecting the design problem we wanted to ensure that the requirements for building an expert system were satisfied. From our previous experience, it seemed unlikely that any single or small group of experts would be sufficient to get reasonable answers to questions about suitability unless we were extraordinarily lucky. Therefore, we devised an approach based on systematically interviewing as large and diverse a group of experts as possible.

Our initial discussions with some of the managers at RBG had identified a promising problem area and two experts who were interested in collaborating with us. We then asked them to arrange for us to interview as many experts in that problem area as possible. We developed the following kinds of guidelines for these interviews.

- First, our resident experts would collect a few typical design problems that they had worked on
- Second, we sought other experts from diverse backgrounds. For example, some might be from advanced development divisions and others from product units. should have worked on different copier designs. Similarly, some might have worked on low-volume copiers versus high-volume copiers.
- Third, the chosen problems should not be familiar to the other experts but be reasonable ones for them to work on

We, in our role as knowledge engineers, then set up the following format for interviewing the experts. We conducted the interviews ourselves but used our chief collaborators as resident experts. The resident experts would present the design problem, and after that essentially act as consultants on the problem but otherwise not participate in the design process. The same problem would be posed to each of the experts in turn, with none of the other experts present in the room. The experts were then asked to carry out the design process, not necessarily doing detailed design, but as much as they could do in roughly two hours. It is important to emphasize that we were *not* asking them to tell us how they solved the design problem but to do the sample design from the given specifications. We believe that experts cannot reliably give an account of their expertise: We have to exercise their expertise on real problems to extract and model their knowledge. Our experience, as described below, provides additional support for this viewpoint.

What was Learned from the Interviews.

Our resident experts had given us the following kinds of information prior to talking to the other experts.

- We asked them to "solve" the design problems the other experts would get to work on. This enabled us to form an initial model of how such designs were carried out in terms of some of the major steps.
- They also identified one of the design sub-problems as being a good candidate for building an expert system
- We ended up examining how five other experts worked on the same problem. The results were both reassuring and surprising.

Commonality of Approach

All the experts followed a very similar strategy in carrying out the design in terms of how they decomposed the problem into sub-problems, worked on the sub-problems, and then related the partial designs to each other. This is not very surprising given that problem decomposition is a time-tested strategy for solving hard problems. What surprised us was the similarity in the sub-problems that were chosen and the strategies used for solving the subproblems. This was very reassuring, in that the selected problem area seemed to have some regularity that each expert was trained to exploit. By the end of the day we could develop an overall protocol for designing the selected sub-system of copying machines. However, we also learned a great deal from the differences in the approaches of the different experts.

Differences in Specialization

Although different experts followed a similar protocol in the design, it became clear that there were sub-areas of expertise in which different experts could specialize. For example, one person was an expert on material selection and knew more about it than others. Another person was an expert on jam clearance strategies, an issue in design that our resident experts had not told us much about. Some of the other areas of specialization that emerged from these interviews were cost analysis, meeting very tight specifications, and using alternate technologies. We found this to be an important observation for a number of reasons.

- As there seemed to be a shared level of expertise, it made sense to try to capture it as an expert system
- There were other layers of specialized knowledge that could be viewed as community knowledge A later effort on our part could try to capture this as a community knowledge base.
- We learned who the different specialists were so that we could talk to them in more depth as the need arose

Nature of Expertise

A related observation was also interesting. While the different experts had a shared understanding of the task domain, there were differences in the nature of their expertise We suspect it may relate to how much the expertise was compiled and in what it was grounded Some experts could carry out the design without thinking very much, even though this was a new problem to them. Others seemed to think more and needed to go back to first principles more often. These latter experts were the ones who had not worked on such problems for some time The difference in grounding came up vividly when we questioned them about why they performed a certain step, why they followed some design rule, or why some constraints were more important than others. The best explanation given by some experts was, "We have always done it this way *and* it seems to work," while others would quote extensively on the theoretical basis for some design decision.

Separability of Task

An early surprise was that the sub-problem identified by our resident experts seemed too contrived. None of the experts really worked on that sub-problem in isolation. It was very dependent on the rest of the design in that it was constrained by some other parts of the design and was in turn constrained by them. The degree of interdependence made it clear that we would have to consider the complete problem or none of it.

Exercising the Expertise

Having different people work on the same problem helped to fill in many holes in the specifications that our resident experts had prepared for presentation to our panel of experts. Often the other experts would ask for information that was not there. Invariably, it turned out that our resident experts had forgotten to include it or had become so used to working with such problems that they could not easily enumerate all the dimensions of the problem- -their expertise needed to be exercised to obtain the knowledge from them This experience was invaluable when we later obtained detailed knowledge from our resident experts.

Discussion

One of the areas of expert systems activity least touched upon in literature is the early part of a project when issues about suitability of a problem and identification of experts are decided. It is our belief that this phase is at least as important as the latter part where technical issues about knowledge representation, problem solving strategies and programming technologies are decided. One of our purposes in this anecdotal article is to shed more light on this early phase and open up some debate and more rigorous experimentation in this regard. One technique that we found useful is the process of systematically interviewing a diverse collection of experts in the tentative problem area and performing some protocol analysis of their problem-solving activity on typical problems. It is our belief that many of the questions that directly affect the outcome of an expert systems project can be answered from interviewing multiple experts We briefly discuss some of these questions.

Identifying Experts and Problems

Who are the experts and what problems do they typically work on? Who is the expert or experts from whom detailed knowledge will be acquired? Are they acknowledged as valuable sources of knowledge among their peers? Talking to a diverse group of experts also helps in determining the standing of the collaborating experts in their community.

Identifying the Users

Do the experts agree on the importance of the given problem, either from the point of view of making their knowledge more widely accessible, or in helping them do a better job? Are they willing to invest resources in the building of the system? It seems to be important to not only identify the clients of the expert system but to judge their commitment in terms of how much they are willing to invest. If they are not willing to invest in it, then perhaps they don't need an expert system (or any other kind for that matter). Experience with the R1 project (McDermott, 1981) is very important in this regard.

Separability of Task Knowledge

Is the given problem a viable one, *i.e.,* can it be reasonably separated from the other activities of the experts, or is it intricately tied to other kinds of knowledge and problem solving? Can you identify the kinds of problem-dependent knowledge and strategies the experts are using or do they seem to rely upon common sense approaches and generic knowledge? This issue is at the heart of whether an expert system is at all possible for some task without having to include all kinds of general reasoning capabilities. Again, by systematically talking to a diverse collection of experts you can make a better judgement on this issue. During this DARN project, one problem area that seemed attractive was troubleshooting problems in computer networks. However, detailed discussions with some of the technicians revealed that because they relied on so many different kinds of general knowledge and reasoning—common sense spatial reasoning, cost and utilities of going between different sites, underlying telephone networks, etc.—that it was not a viable project, at least in the short term. This issue also seems to be related to the compilation of task-specific knowledge (Chandrasekaran and Mittal, 1983a; Riesback, 1984) from more fundamental knowledge.

What Knowledge Will Be Captured in the Expert System?

Is there some commonality in how the different experts solve the problems, or does each expert follow a different, idiosyncratic approach? If there are different approaches, are they equally valid? Are the differences caused by specializations among experts for different sub-problems? Does the expertise for solving the selected problem reside in individuals or is it a community involvement? These are key issues about the nature of the expertise needed in an expert system. Different approaches seem to have been taken for resolving these issues. The typical approach is based on identifying certain key expert(s) in their field and basing the system on their knowledge and experience The INTERNIST project seems to have been quite successful with this approach, which finesses many of the issues by appealing to some established authority, and in fact may be a reasonable approach for tasks where there are clearly acknowledged experts.

A different approach is based on capturing a community knowledge with different experts contributing knowledge suitable for different aspects of the problem. Some (Stefik and Conway, 1982) have even argued that in many domains, such as VLSI design, conventional means of acquiring knowledge from experts may not even be feasible because such knowledge may be too diffusely spread among different experts and subject to rapid changes. In such cases, the knowledge base itself may need to be designed. In any case, much research needs to be done to resolve issues such as identifying different aspects of a problem and corresponding experts, integrating knowledge from various experts, resolving conflicts, assimilating competing strategies, personalizing community knowledge bases, and developing programming technologies for supporting these activities.

We envision that the PRIDE system will make available a level of expertise that is better than any single expert's in the following sense It will become a community knowledge base which will not only have methods and a terminology that is practiced in common by all the design experts, but also capture some of the areas in which these experts specialize.

Conclusion

We have identified some of the problems in selecting suitable tasks for expert system projects and acquiring knowledge from experts. In the PRIDE project, our experience with using multiple experts to understand and define the task has been quite successful. The initial protocol of the design process, obtained from many experts, has proved to be quite stable despite many iterative refinements. As we get more experience with operational versions of the systems, we expect to gain a better understanding of the differences across different experts.

References

Chandrasekaran, B & S Mittal (1983) Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems In M C Yovits, (Ed.,) *Advances in Computers,* Vol 22 New York: Academic Press, pp 218-295

Chandrasekaran, B. & S. Mittal (1983) *Deep versus compiled knowledge approaches to diagnostic problem solving.* AAAI-82, pp. 349-354. (Revised version in *International Journal of Man-Machine Studies* 19, pp. 425-436

Chandrasekaran, B., S Mittal, & J W. Smith, (1980) RADEX—Towards a Computer-Based Radiology Consultant. In E S Gelsema & L. N Kanal, (Eds ,) *Pattern Recognition in Practice* Amsterdam: North-Holland, pp. 463-474

Dixon, J R & M K Simmons. (1983) Computers That Design: Expert Systems for Mechanical Engineers *Computers in Mechanical Engineering,* pp 10-18

Hayes-Roth, F , D. A. Waterman, & D. B Lenat, (Eds ,) *Building Expert Systems* Reading, Mass: Addison-Wesley

McDermott, J. (1981) R1: The Formative Years *AI Magazine,* Vol 2, No 2, pp 21-29

Mittal, S. Representation and Execution of Design Plans in PRIDE In preparation

Mittal, S , D G Bobrow, & J de Kleer (1984) DARN: A Community Memory for a Diagnosis and Repair Task Unpublished report, Xerox PARC, Palo Alto, CA

Mittal, S., B Chandrasekaran, & J Sticklen. (1984) PATREC: A Knowledge-Directed Database for a Diagnostic Expert System *Computer,* Vol. 19, No 9, pp 51-59.

Mittal, S , M. Morjaria, & C L Dym (1985) Overview of PRIDE: An Expert System for the Design of Paper Handling Systems in Copiers To be presented at the 1985 ASME Winter Annual Meeting, November, 1985

Pople, H. (1977) *The formation of composite hypotheses in diagnostic problem solving: an exercise in synthetic reasoning* IJCAI-5, pp 1030-1037

Riesback, C K (1984) Knowledge Reorganization & Reasoning Style *International Journal of Man-Machine Studies,* 20: pp 45-61

Smith, R G. & J D Baker (1983) *The dipmeter advisor system—a case study in commercial expert system development* IJCAI-8, pp 122-129

Stefik, M J & L Conway (1982) Towards a Principled Engineering of Knowledge, *AI Magazine,* Vol 3, No 3, pp 4-16

Tatman, J L & J. W Smith (1982) Preliminary Evaluation of MDX - A Medical Diagnosis System 6th Annual Symp. Computer Applicaton Medical Care Washington D C